

Package ‘CMLbeta’

August 12, 2019

Type Package

Title Conditional Maximum likelihood for two-phase designs

Version 0.1.0

Author Claudia Rivera Rodriguez

Maintainer The package maintainer <c.rodriguez@auckland.ac.nz>

Description Conditional Maximum likelihood for two-phase designs, including gaussian, binomial, poisson and Gamma models.

License What license is it under?

Encoding UTF-8

LazyData true

R topics documented:

CMLcal	1
CMLest	4
Index	8

CMLcal	<i>Conditional maximum likelihood and Weighted likelihood for two-phase designs with calibrated weights</i>
--------	---

Description

Fits a generalize linear model to data arising from two phase designs using CML and WL with calibrated weights

Usage

```
CMLcal(ids=1:length(Ya), Ya, pi0a,bet = NULL, Za, dist='gaussian',
link='identity', R.PI, X.PINoy , distance='chi', eps=1E-10,
bet.PI =NULL, fX.PI=formula(~ 1), pFUN0 ,X.PI.TRUENoy)
```

Arguments

ids	A vector of unique identifiers. ids
Ya	The variable outcome for the first-phase sample. The outcome should be available for all subjects at first phase.
pi0a	A vector of sampling inclusion probabilities for all subjects at first phase
bet	starting values of the parameters
Za	A matrix of model covariates for all first phase subjets, missing values must be set to NA
dist	Distribution of the outcome: 'gaussian', 'binomial', 'poisson' or 'Gamma'.
link	Link function: 'identity', 'log', 'inverse' (for gaussian). 'logit', 'log', 'probit' (for binomial). 'log', 'sqrt' (for poisson). 'inverse2', 'log' (for Gamma)
R.PI	A vector of sampling indicators
X.PINoy	A matrix of data available at first phase included in the analysis, excluding the outcome distance='chi', eps=1E-10, bet.PI=NULL, fX.PI=formula(~ 1), pFUN0,X.P.TRUENoy)
distance	A distance option for calibrated weights- options are 'chi' and 'poisson' (same as raking).
eps	Maximum error of Newton Raphson algorithm
bet.PI	Starting values for the model of inclusion probabilities
fX.PI	Formula to model the inclusion probabilities- e.g formula(~Ya) or formula(~Ya+X1)
pFUN0	A function specifying the original sampling probabilities, e.g $f(x,y) 1/(1+\exp(-1-2x-2y))$
X.PI.TRUENoy	A matrix of data available at first phase used for the sampling design, excluding the outcome. Usually X.PI.TRUENoy is a subset of X.PINoy

Details

CMLCal is a function for CML with calibrated sampling probabilities. .

Value

CMLcal() returns a list that includes estimated coefficients and an estimate of their asymptotic covariance matrix:

coeffWL	Weighted likelihood(WL) estimated coefficients
coeffCL	Conditional likelihood(CML) estimated coefficients
alp	Coefficient for the calibrated weights with WL
alp.c	Coefficient for the calibrated weights with CML
covT	Covariance matrix for the coefficients using WL, including the coefficients for the calibrated weights
covT.c	Covariance matrix for the coefficients using CML, including the coefficients for the calibrated weights
eps	error in the last iteration of algorithm

Author(s)

Claudia Rivera-Rodriguez

References

Scott, Alastair J., and Chris J. WILD. "Fitting Regression Models with Response-Biased Samples." *The Canadian Journal of Statistics / La Revue Canadienne De Statistique*, vol. 39, no. 3, 2011, pp. 519–536.

Rivera-Rodriguez, Claudia and Haneuse, Sebastien & Wang, Molin and Spiegelman, Donna. (2019). Augmented pseudo-likelihood estimation for two-phase studies. *Statistical Methods in Medical Research*.

Examples

```
set.seed(1093183)

N <- 5000
n<-500
Za <- abs(cbind(1, rnorm(N), rnorm(N), rbinom(N, 1, 0.3)))
Xa <- cbind(1, Za[,c( 4)])

## REsponse variables
bet <- c(1.5,1,1,1)
YaNormal <- rnorm(N, ( Za%*%bet), 1); mean(YaNormal)

Ya <- YaNormal
bet.PI <- c(-1,-0.5)
pi0a <- 1/(1+exp(-cbind(1,Ya)%*%bet.PI )); sum(pi0a) ;
R <- R.PI <- rbinom(N, 1, pi0a) ; sum(R)
pi0M <- glm(R~ Ya , family=binomial )

X.PI.TRUE <- cbind( 1, Ya )
X.PI.TRUE.Noy <- cbind( Xa[,1] )
colnames(X.PI.TRUE) <- c('Int', 'Ya')
colnames(X.PI.TRUE.Noy) <- c('Int' )
bet.PITRUE <- pi0M$coefficients
## Formula for the sampling weights :
fX.PI0 <- formula(~0+ Int+ Ya )
## Fomula and covariates for the calibrated/estimated weights:
X.PINoy <- cbind( Xa )
colnames(X.PINoy) <- c('Int', 'x2' )
fX.PI <- formula(~0+ Int+x2*Ya )

## Epsilon for estimation process:
eps=1E-10
ids=ids.PI=1:length(R.PI)
## Distribution and link for ESTIMATED WEIGHTS
dist.PI='binomial'; link.PI='logit'

## Distribution and link for OUTCOME Ya
dist= 'gaussian' ; link='identity'
ids.PI=1:length(Ya)
simulta=FALSE
bet.PI= bet.PITRUE
```

```

bet.PIy <- bet.PITRUE[2]
bet.PIx <- bet.PITRUE[1]

pFUN0 <- function(x.pi, yi, ...){
  x <- cbind(rbind(x.pi) , yi)
  b <- c(bet.PIx, bet.PIy )
  #x <- cbind(rbind(x.pi) )
  #b <- c(bet.PIx )
  pii0 <- 1/(1+exp( -x*%b ))
  pii0}

# This function allows interaction in the weights

bet <- bet
distance<- 'poisson'
modSc_AlpCal <- CMLcal( ids=1:length(Ya), Ya, pi0a, bet =bet, Za, dist=dist, link=link,
                      R.PI, X.PINoy , distance=distance, eps=1E-10, bet.PI =NULL, fX.PI=formula(~ 1), pFUN0

## Conditional Max. Lik. calibration
modSc_AlpCal$coeffCL

# Weighted Lik. calibration
modSc_AlpCal$coeffWL

```

CMLest

Conditional maximum likelihood and Weighted likelihood for two-phase designs with estimated weights

Description

Fits a generalize linear model to data arising from two phase designs using CML and WL with estimated weights

Usage

```

CMLest(ids=1:length(Ya), Ya, pi0a,bet = NULL, Za, dist='gaussian',
link='identity', R.PI, X.PINoy, dist.PI='binomial', link.PI='identity' ,
eps=1E-10 , bet.PI =NULL, fX.PI = fX.PI )

```

Arguments

<code>ids</code>	A vector of unique identifiers. <code>ids</code>
<code>Ya</code>	The variable outcome for the first-phase sample. The outcome should be available for all subjects at first phase.
<code>pi0a</code>	A vector of sampling inclusion probabilities for all subjects at first phase
<code>bet</code>	starting values of the parameters
<code>Za</code>	A matrix of model covariates for all first phase subjects, missing values must be set to NA

dist	Distribution of the outcome: 'gaussian', 'binomial', 'poisson' or 'Gamma'.
link	Link function: 'identity', 'log', 'inverse' (for gaussian). 'logit', 'log', 'probit' (for binomial). 'log', 'sqrt' (for poisson). 'inverse2', 'log' (for Gamma)
R.PI	A vector of sampling indicators
X.PI _{Noy}	A matrix of data available at first phase included in the analysis, excluding the outcome
dist.PI	Distribution to model the inclusion probabilities- usually binomial
link.PI	Link for the model of the inclusion probabilities (logit, log or probit for dist.PI='binomial')
eps	Maximum error of Newton Raphson algorithm
bet.PI	Starting values for the model of inclusion probabilities
fX.PI	Formula to model the inclusion probabilities- e.g formula(~Ya) or formula(~Ya+ X1)

Details

CMLest is a function for CML. For logistic regression this method is equivalent to Scot & Wild (2011).

Value

CMLest() returns a list that includes estimated coefficients and an estimate of their asymptotic covariance matrix:

coeffWL	Weighted likelihood(WL) estimated coefficients
coeffCL	Conditional likelihood(CML) estimated coefficients
alp	Coefficient for the estimated weights with WL
alp.c	Coefficient for the estimated weights with CML
covT	Covariance matrix for the coefficients using WL, including the coefficients for the estimated weights
covT.c	Covariance matrix for the coefficients using CML, including the coefficients for the estimated weights
eps	error in the last iteration of algorithm

Author(s)

Claudia Rivera-Rodriguez

References

- Scott, Alastair J., and Chris J. WILD. "Fitting Regression Models with Response-Biased Samples." *The Canadian Journal of Statistics / La Revue Canadienne De Statistique*, vol. 39, no. 3, 2011, pp. 519–536.
- Rivera-Rodriguez, Claudia and Haneuse, Sebastien and Wang, Molin and Spiegelman, Donna. (2019). Augmented pseudo-likelihood estimation for two-phase studies. *Statistical Methods in Medical Research*.

Examples

```

# NORMAL MODEL
set.seed(1093183)

N <- 5000
n<-500
Za <- abs(cbind(1, rnorm(N), rnorm(N), rbinom(N, 1, 0.3)))
Xa <- cbind(1, Za[,c( 4)])

## REsponse variables
bet <- c(1.5,1,1,1)
YaNormal <- rnorm(N, ( Za%*%bet), 1); mean(YaNormal)

Ya <- YaNormal
bet.PI <- c(-1,-0.5)
pi0a <- 1/(1+exp(-cbind(1,Ya)%*%bet.PI )); sum(pi0a) ;
R <- R.PI <- rbinom(N, 1, pi0a ) ; sum(R)
pi0M <- glm(R~ Ya , family=binomial )

X.PI.TRUE <- cbind( 1, Ya )
X.PI.TRUENoy <- cbind( Xa[,1] )
colnames(X.PI.TRUE) <- c('Int', 'Ya')
colnames(X.PI.TRUENoy) <- c('Int' )
bet.PITRUE <- pi0M$coefficients
## Formula for the sampling weights :
fX.PI0 <- formula(~0+ Int+ Ya )
## Fomula and covariates for the calibrated/estimated weights:
X.PINoy <- cbind( Xa )
colnames(X.PINoy) <- c('Int', 'x2' )
fX.PI <- formula(~0+ Int+x2*Ya )

## Epsilon for estimation process:
eps=1E-10
ids=ids.PI=1:length(R.PI)
## Distribution and link for ESTIMATED WEIGHTS
dist.PI='binomial'; link.PI='logit'

## Distribution and link for OUTCOME Ya
dist= 'gaussian' ; link='identity'
ids.PI=1:length(Ya)
simulta=FALSE
bet.PI= bet.PITRUE

modSc_AlpEst <- CMLest( ids=1:length(Ya), Ya, pi0a,bet = bet, Za, dist=dist, link=link,
R.PI, X.PINoy, dist.PI=dist.PI, link.PI=link.PI , eps=1E-10 , bet.PI =NULL, fX.PI = fX.PI )

## Conditional Max. Lik. estimated weg
modSc_AlpEst$coeffCL

```

```
# Weighted Lik. estimated weg  
modSc_Alpest$coeffWL
```

Index

CMLcal, [1](#)
CMLest, [4](#)